Trajectory Optimization for Wheeled-Legged Quadrupedal Robots using Linearized ZMP Constraints

Journal Article

Author(s):
de Viragh, Yvain; Bjelonic, Marko; Bellicoso, C. Dario; Jenelten, Fabian; Hutter, Marco (iD)

Trajectory Optimization for Wheeled-Legged Quadrupedal Robots using Linearized ZMP Constraints

Yvain de Viragh, Marko Bjelonic, C. Dario Bellicoso, Fabian Jenelten, and Marco Hutter

Abstract—We present a trajectory optimizer for quadrupedal robots with actuated wheels. By solving for angular, vertical, and planar components of the base and feet trajectories in a cascaded fashion and by introducing a novel linear formulation of the zero-moment point (ZMP) balance criterion, we rely on quadratic programming only, thereby eliminating the need for nonlinear optimization routines. Yet, even for gaits containing full flight phases, we are able to generate trajectories for executing complex motions that involve simultaneous driving, walking, and turning. We verified our approach in simulations of the quadrupedal robot ANYmal equipped with wheels, where we are able to run the proposed trajectory optimizer at 50 Hz. To the best of our knowledge, this is the first time that such dynamic motions are demonstrated for wheeled-legged quadrupedal robots using an online motion planner.

Index Terms—Legged Robots, Wheeled Robots, Motion and Path Planning, Optimization and Optimal Control

I. INTRODUCTION

WHEELED-LEGGED robots offer the potential to combine the best of two locomotion domains: The efficiency and speed of wheels with the ability of legged robots to cope with challenging terrain. Search and rescue, where time can be a matter of life and death, is only one example of tasks that could greatly benefit from such systems. However, so far no locomotion framework has been published that would allow to fully exploit one of the main advantages of this combination, that is, the ability to simultaneously walk and drive. This work bridges this gap by presenting a trajectory optimizer (TO) that finds online highly dynamic motions for hybrid[1] locomotion of wheeled-legged quadrupedal robots, such as the one shown in Fig. 1, that has no wheel steering. The ability to cope with the latter is attractive from an engineering point of view in the sense that it allows reducing mechanical complexity, total weight, and leg inertia in favor of reliability, energy consumption, and agility, respectively.

[1]In the context of this work, we use "hybrid" as an attribute that denotes simultaneous walking and driving.
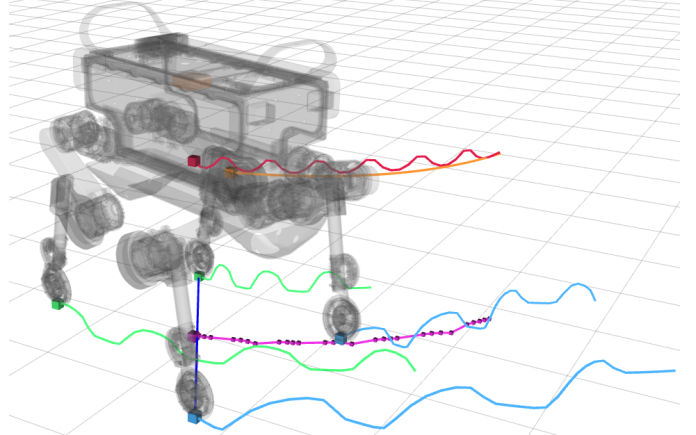

Fig. 1. Gazebo simulation of ANYmal [1] equipped with actuated, non-steerable wheels executing a hybrid, full flight trotting gait while tracking a reference trajectory (orange) that includes a 90° turn. The center of mass (CoM) and wheel trajectories (shown in red, green, and blue) for this highly dynamic motion were generated by our trajectory optimizer (TO). A video demonstrating many more can be found at https://youtu.be/I1aTCTc0J4U.

A. Related Work

Finding input and state trajectories for legged robots using a model-based approach is an involved problem due to the high nonlinearity of the system dynamics. To obtain a problem that can be solved online, three techniques are commonly employed: 1) introducing assumptions to reduce model complexity 2) solving the problem in a hierarchical cascade such that a set of simpler subproblems is obtained, and 3) using heuristics to obtain parts of the solution.

An example using all three techniques is the policy-regularized model predictive controller (MPC) for the quadrupedal robot *MIT Cheetah* [2]. Their robot model is a floating base with massless legs and quadratic velocity terms neglected. The swing timings of the legs are found a priori and a heuristic based on Raibert's foothold prediction [3] is used as an objective to guide the optimization search direction. The reported solver times are in the order of 100 ms for a planning horizon of one stride duration[2].

A TO that also optimizes over the swing timings is presented in [4]. Again, the robot model is a floating base with massless legs. Heuristics enter in the form of the initial guess and the prefixed number of steps. The resulting solver times for a planning horizon of one stride duration were stated as 100 ms [5].

[2]The authors used a MATLAB implementation, while all other works mentioned rely on C++.

Previous work of Winkler et al. showed a TO formulation which finds the planar CoM and feet trajectories using nonlinear programming (NLP) [6]. The swing timings are determined a priori, and the robot is modeled as a linear inverted pendulum, with constant height and fixed orientation. For a planning horizon of one stride duration, these drastic simplifications resulted in solver times between 10 and 120 ms, depending on the gait.

If the feet trajectories are computed first using heuristics such as Raibert's foothold prediction, the CoM trajectory can be obtained by quadratic programming (QP)[3] and solver times decrease to a few milliseconds [7]. The authors have recently generalized their approach to include the vertical CoM motion into an NLP problem [8]. This allows to find trajectories for gaits containing full flight phases, but also increased solver times more than fivefold.

The model simplifications in [6]–[8] rely to a large extent on the so-called zero-moment point (ZMP) balance criterion. It allows for considerable abstractions by formulating a condition for dynamic balancing that relates the stance feet positions to the linear and angular momentum of the system. Its application has a long history in legged robotics [9]. However, it is important to realize that the associated gain in computational efficiency comes at the cost of reduced generality. For instance, the extension to uneven terrain [10] and the inclusion of friction cone constraints is not straightforward – a drawback from which computationally more involved methods such as [4] do not suffer.

As illustrated by above works, there has been a large interest in generating dynamic motions for conventional legged robots. By contrast, research on wheeled-legged robots [11]–[21] has typically focused on statically stable locomotion over uneven terrain. Examples include the space exploration vehicles [11]–[13], which use their limbs rather as sophisticated suspension systems than as legs with wheels as end-effectors. The approach to traverse flat terrain by driving and irregular terrain by walking was further explored in [14]–[17]. However, the possibilities offered by simultaneous walking and driving were not much investigated, even for statically stable motions. In this regard, two important exceptions are *Boston Dynamic*'s wheeled biped *Handle* [22] and the recent contribution [23], which has shown a generic approach to motion generation for wheeled-legged robots based on NLP. However, little is known about Handle's control framework, and the computation times of [23] are typically a multiple of the planning horizon, making it prohibitively slow for online application in its present form.

B. Contribution

We present a TO for quadrupedal robots with actuated wheels that finds trajectories for walking, driving, and hybrid combinations thereof, given the gait pattern and the desired goal state. We solve for the planar components of the CoM and feet trajectories in a single optimization and, in contrast to [6]–[8], we compute the angular components and the vertical
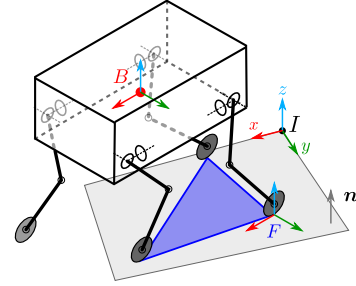

Fig. 2. Model of a wheeled-legged quadrupedal robot with massless legs and planar, nondeformable wheels. The current support polygon is shown in shaded blue. I denotes an inertial frame with z-axis collinear to the ground plane normal $\boldsymbol{n}$, and B the base frame with origin at the robot's CoM. We let the frame F be fixed at a leg's endpoint, i.e., the point that during stance is in contact with the ground (shown for the left hind (LH) leg only), and define this point as a leg's foot. This is a useful definition for our case, as we can model conventional point-contact feet and wheels simply by changing the kinematic constraints at F. Namely, by defining the z-axis of F to be aligned with the plane normal and the x-axis to be perpendicular to the wheel's rotation axis, the difference between the two becomes only whether F may have a non-zero velocity component along its x-direction.

CoM motion prior to the optimization. This allows us to find trajectories for driving curves and executing gaits with full flight phases without solving an NLP problem. Further, we introduce a parameterization of the stance feet trajectories that inherently fulfills the kinematic rolling constraint. To the best of our knowledge, we present a novel way of formulating the ZMP balance criterion as a linear constraint.

As we demonstrate in simulations, this allows us to generate trajectories for dynamic, hybrid locomotion of wheeled-legged quadrupedal robots with unprecedented speed. We complete the presentation of our TO by a discussion on its applicability to real systems and irregular terrain.

II. PROBLEM FORMULATION

For our TO we assume a simplified robot model where the legs are massless. The single floating body has the total mass of the robot and an inertia that corresponds to a default configuration. Furthermore, we model the wheels as flat, nondeformable disks.

Fig. 2 illustrates our definitions of the inertial, base, and foot frames I, B, and F, respectively. In the case of a wheeled foot, the motion of F is subject to the kinematic rolling constraint during stance. As we solve for the planar base and feet trajectories simultaneously in our TO, but seek to avoid nonlinear constraints, we need the following condition to hold: The kinematic constraints on F must not depend on its relative position to the base. For a conventional point-contact foot, this is trivially satisfied, as it is fixed during stance. However, in case of a wheel, it imposes restrictions on the motion. For ANYmal [1] equipped with non-steerable wheels, the condition is satisfied if the base pitch angle is zero with respect to (w.r.t.) the inertial frame. This is outlined further in the Appendix.

A. Cascaded Trajectory Generation

Fig. 3 shows the complete architecture of our TO. The high-level inputs are the gait pattern[4] and a reference goal

---

[3]Strictly speaking, QP is a subset of NLP. However, we here use the term NLP to explicitly denote optimization problems that contain constraints that are not linear, and can thus not be solved using QP.

[4]The gait pattern defines the swing and stance timings of each leg, i.e., when the leg's endpoint F should be in contact with the ground.
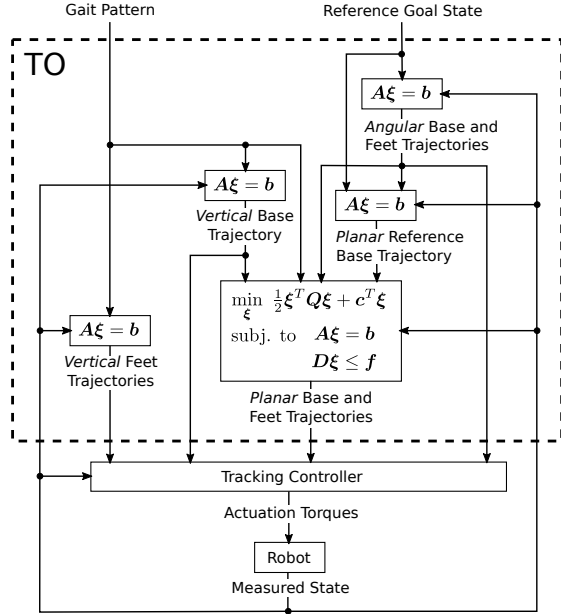
Fig. 3. Architecture of our cascaded TO. As explained in Section III, we are able to generate all trajectories by solving systems of linear equations (denoted as $\boldsymbol{A\xi} = \boldsymbol{b}$) or, in the case of the planar base and feet trajectories, a single QP problem. Note that, instead of a reference goal state, one may also directly provide the angular trajectories and the planar reference base trajectory.

state consisting of linear and angular base position and/or velocity. If the gait pattern contains full flight phases, i.e., all legs are in swing phase, a time-varying trajectory for the z-component of the base is computed that accounts for these ballistic free-fall phases. The goal state is used to compute the angular trajectories and a planar reference base trajectory. Alternatively, these can also directly be provided, for instance by a high-level path planner. As the vertical foot motion does not affect the optimization, we compute it separately, whereby we parameterize each swing phase by a quintic polynomial. To track the trajectories and compute actuation torques, a whole-body controller (WBC), such as the one described in [24], can for instance be used.

B. Linearized ZMP Balance Criterion

One of our main contributions consists of a linear formulation of the ZMP balance criterion. We derive it starting from our simplified robot model. Namely, assuming mass-less legs, the dynamics of the robot are those of a free-floating rigid body given by

$$\begin{bmatrix} m \, _I\boldsymbol{a}_{IB} - m \, _I\boldsymbol{g} \\ _B\boldsymbol{\Theta}_{BB}\dot{\boldsymbol{\omega}}_{IB} + _B\boldsymbol{\omega}_{IB} \times _B\boldsymbol{\Theta}_{BB}\boldsymbol{\omega}_{IB} \end{bmatrix} = \begin{bmatrix} _I\boldsymbol{F}_{\text{ext}} \\ _B\boldsymbol{\Gamma}_{\text{ext}} \end{bmatrix}, \quad (1)$$

where m denotes the robot's mass, $\boldsymbol{\Theta}_B$ its inertia tensor, $\boldsymbol{g}$ the gravity vector, and $\boldsymbol{F}_{\text{ext}}$, $\boldsymbol{\Gamma}_{\text{ext}}$ the external forces and torques acting at the CoM. The left-hand subscripts specify in which frame the quantities are expressed. We let the symbols $\boldsymbol{a}$, $\boldsymbol{v}$, $\boldsymbol{r}$, and $\boldsymbol{\omega}$ denote linear acceleration, velocity, position, and angular velocity, respectively. As an example, $_I\boldsymbol{r}_{BF}$ is the position vector of F w.r.t. B expressed in I. For a flat ground plane, the ZMP is defined as the point on the ground where the moment induced by the gravito-inertial forces – the left-hand
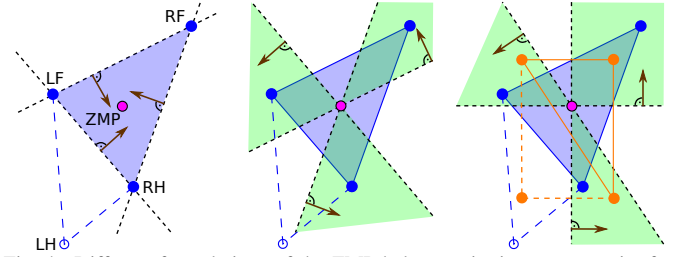

Fig. 4. Different formulations of the ZMP balance criterion as constraint for the case of a three-leg support phase where the left fore (LF), right fore (RF), and right hind (RH) leg are in stance phase and the left hind (LH) leg is swinging. Left image: Constraint on the ZMP (magenta circle) to lie within the current support polygon (blue shaded area). Middle image: An equivalent formulation where the constraint edges (dotted lines) have been shifted such as to intersect at the ZMP and where the feet (blue circles) are constrained to lie in the associated convex cones. In other words, the edge directions (represented by normal vectors in brown) are those of the lines connecting the feet. This formulation leads to doubling of the constraint number. Right image: Linear constraints are obtained by fixing the constraint edge directions a priori. If the angular motion of the base is known, this can, for instance, be done by choosing them as the directions of the lines connecting the hips (orange).

side (LHS) of (1) – has only a component in the direction of the plane normal $\boldsymbol{n}$. I.e., it must hold that

$$_I\boldsymbol{n} \times \left(_I\boldsymbol{r}_{IB} - _I\boldsymbol{r}_{I\text{ZMP}} \right) \times \left(m \, _I\boldsymbol{a}_{IB} - m \, _I\boldsymbol{g} \right)$$
$$= _I\boldsymbol{n} \times \boldsymbol{R}_{IB} \left(_B\boldsymbol{\Theta}_{BB}\dot{\boldsymbol{\omega}}_{IB} + _B\boldsymbol{\omega}_{IB} \times _B\boldsymbol{\Theta}_{BB}\boldsymbol{\omega}_{IB} \right), \quad (2)$$

where $\boldsymbol{R}_{IB}$ is the passive rotation matrix from the inertial frame to the base frame such that $_I\boldsymbol{r} = \boldsymbol{R}_{IB} \, _B\boldsymbol{r}$. As we let the inertial frame's z-axis coincide with the plane normal, (2) can be solved for the x,y-components of the ZMP as

$$_I\boldsymbol{r}_{I\text{ZMP}}^{xy} = _I\boldsymbol{r}_{IB}^{xy} - \frac{1}{\ddot{z}_B - g_z} \left(z_B \left(_I\boldsymbol{a}_{IB}^{xy} - _I\boldsymbol{g}^{xy} \right) + \boldsymbol{L}^{xy} \right), \quad (3)$$

where z_B and g_z are the z-component of $_I\boldsymbol{r}_{IB}$ and $_I\boldsymbol{g}$, respectively, and $(\cdot)^{xy}$ denotes a quantity's planar components. The angular contributions are given by

$$\boldsymbol{L}^{xy} = \begin{bmatrix} -\boldsymbol{R}_{IB,2} \\ \boldsymbol{R}_{IB,1} \end{bmatrix} \left(_B\boldsymbol{\Theta}_{BB}\dot{\boldsymbol{\omega}}_{IB} + _B\boldsymbol{\omega}_{IB} \times _B\boldsymbol{\Theta}_{BB}\boldsymbol{\omega}_{IB} \right), \quad (4)$$

where $\boldsymbol{R}_{IB,i}$ denotes the i-th row of $\boldsymbol{R}_{IB}$. Note that for z_B constant, (3) represents the equations of motion (EoM) of a linear inverted pendulum with pivot at the ZMP. For dynamic balancing, the planned ZMP must always lie within the support polygon [10]. In an optimization, this criterion takes the form of polyhedral constraints. These are nonlinear when solving simultaneously for the feet and base trajectories, due to the constraint edge directions depending either on the feet positions or on auxiliary optimization variables, such as in [6]. To obtain a linear constraint, the edge directions must thus be fixed a priori. We propose to do this as illustrated in Fig. 4 for the case of a three-leg support phase. Namely, we require the feet to lie in the convex cones spanned by a set of lines with fixed directions that intersect at the ZMP, such that the resulting support polygon always contains the ZMP. The main drawback of this linearization is that it restricts the optimization variables to lie in a subset of the ones that satisfy the ZMP balance criterion, thus requiring careful selection of the edge directions to obtain a feasible optimization problem.

Intuitively speaking, this subset is characterized by the loss of one rotational degree of freedom (DoF) per edge.

We found that it works well to select the edge directions as the ones that result from connecting the hip positions of the stance legs at touchdown. That is, we start with a set of six direction vectors – one for each combination of feet pairs – and whenever a foot switches from swing to stance, we recompute the associated directions by connecting that foot's hip position with the hip positions of all other stance legs. This procedure ensures that for pairs of grounded feet the directions remain constant w.r.t. the inertial frame during nonzero yaw motions. Otherwise, in the case of pure walking with gaits containing two-leg support phases, the problem would become infeasible. For the same reason, we initialize the edge directions of initially grounded feet pairs by connecting the measured feet positions instead of the hips.

III. TRAJECTORY OPTIMIZATION

We formulate the task of finding the x,y-components of the feet and base trajectories as a QP problem of the form

$$\min_{\boldsymbol{\xi}} \frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{Q}\boldsymbol{\xi} + \boldsymbol{c}^T \boldsymbol{\xi}, \quad \text{subj. to } \boldsymbol{A}\boldsymbol{\xi} = \boldsymbol{b}, \ \boldsymbol{D}\boldsymbol{\xi} \leq \boldsymbol{f}, \quad (5)$$

where the variables parameterizing the trajectories are stacked in the vector of optimization variables $\boldsymbol{\xi}$. In the following, this parameterization is presented in more detail, and we introduce the objectives and constraints that contribute to the QP problem. Furthermore, we propose a warm-starting scheme and briefly discuss how the remaining trajectories, which serve as inputs to the QP problem, can be generated by solving systems of linear equations.

A. Parameterization

We parameterize the planar trajectories by sequences of connected polynomials. We write an n-th order polynomial starting at time t^{start} as

$$p(t) = \boldsymbol{\eta}_n(t, t^{\text{start}})^T \boldsymbol{\alpha}, \quad (6)$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{n+1}$ is the vector of coefficients and

$$\boldsymbol{\eta}_n(t, t^{\text{start}}) = \begin{bmatrix} \tau^n & \tau^{n-1} & \dots & \tau & 1 \end{bmatrix}^T \in \mathbb{R}^{n+1}, \quad (7)$$

with $\tau := t - t^{\text{start}}$. Derivatives w.r.t. time are obtained by differentiation of $\boldsymbol{\eta}_n(t, t^{\text{start}})$, e.g., $\dot{p}(t) = \dot{\boldsymbol{\eta}}_n(t, t^{\text{start}})^T \boldsymbol{\alpha}$.

1) Base segments: We compose each base segment of two quintic polynomials describing the planar components of the base position:

$$_I\boldsymbol{r}_{IB}^{xy}(t) = \underbrace{\begin{bmatrix} \boldsymbol{\eta}_5(t, t_{B,k}^{\text{start}})^T & \boldsymbol{0}_{1\times 6} \\ \boldsymbol{0}_{1\times 6} & \boldsymbol{\eta}_5(t, t_{B,k}^{\text{start}})^T \end{bmatrix}}_{:=\boldsymbol{T}_B(t, t_{B,k}^{\text{start}})} \underbrace{\begin{bmatrix} \boldsymbol{\alpha}_{B,k,x} \\ \boldsymbol{\alpha}_{B,k,y} \end{bmatrix}}_{:=\boldsymbol{s}_{B,k}}, \quad (8)$$

where $t_{B,k}^{\text{start}}$ is the start time of the k-th base segment.

2) Swing foot segments: Similarly, we compose segments describing the motion of a foot in swing using pairs of cubic polynomials:

$$_I\boldsymbol{r}_{IF}^{xy}(t) = \underbrace{\begin{bmatrix} \boldsymbol{\eta}_3(t, t_{F,l}^{\text{start}})^T & \boldsymbol{0}_{1\times 4} \\ \boldsymbol{0}_{1\times 4} & \boldsymbol{\eta}_3(t, t_{F,l}^{\text{start}})^T \end{bmatrix}}_{:=\boldsymbol{T}_{F,l}(t, t_{F,l}^{\text{start}})} \underbrace{\begin{bmatrix} \boldsymbol{\alpha}_{F,l,x} \\ \boldsymbol{\alpha}_{F,l,y} \end{bmatrix}}_{:=\boldsymbol{s}_{F,l}}, \quad (9)$$

where $t_{F,l}^{\text{start}}$ is the start time of the l-th foot segment.

3) Stance foot segments: For a wheeled foot, the trajectories must satisfy the rolling constraint in stance phase. That is, the instantaneous velocity must have no component along the y-axis of the foot frame F. We thus parameterize such segments by quadratic polynomials describing the velocity of F along its x-axis and the position $_I\boldsymbol{r}_{IF,l}^{\text{start}}$ at the beginning of the segment. That is, the velocity of F in the inertial frame is given by[5]

$$_I\boldsymbol{v}_{IF}^{xy}(t) = \boldsymbol{R}_{IF}^{xy}(t) \underbrace{\begin{bmatrix} \boldsymbol{\eta}_2(t, t_{F,l}^{\text{start}})^T & 0 & 0 \\ & \boldsymbol{0}_{1\times 5} & \end{bmatrix}}_{:=\dot{\boldsymbol{T}}_{F,l}(t, t_{F,l}^{\text{start}})} \underbrace{\begin{bmatrix} \boldsymbol{\alpha}_{F,l} \\ _I\boldsymbol{r}_{IF,l}^{\text{start},xy} \end{bmatrix}}_{:=\boldsymbol{s}_{F,l}}, \quad (10)$$

and the position is obtained by integration as

$$_I\boldsymbol{r}_{IF}^{xy}(t) = \underbrace{\left(\begin{bmatrix} \boldsymbol{0}_{2\times 3} & \boldsymbol{I} \end{bmatrix} + \int_{t_{F,l}^{\text{start}}}^{t} \dot{\boldsymbol{T}}_{F,l}(\tilde{t}, t_{F,l}^{\text{start}})\, d\tilde{t} \right)}_{=\boldsymbol{T}_{F,l}(t, t_{F,l}^{\text{start}})} \boldsymbol{s}_{F,l},$$

$$(11)$$

where $\boldsymbol{I} \in \mathbb{R}^{2\times 2}$ denotes the identity matrix. Since (11) has no analytic solution for nontrivial angular motions, a numerical integration scheme must be used. However, the complexity of the QP problem is not increased, as the integral does not depend on any optimization variables.

4) Segment durations: We set the solution complexity indirectly through upper bounds on the maximal durations of base, swing foot, and stance foot segments. Given these bounds, we divide each swing and stance phase into a number of equally spaced swing and stance foot segments, respectively, and choose the base segments such that whenever a foot switches between stance and swing, there is also a new base segment starting. This is important for transitions between disjoint support polygons, see Section III-C2.

B. Objectives

The following list gives an overview of the cost terms contributing to the objective function in (5). In order for the resulting Hessian $\boldsymbol{Q}$ to be positive definite, we add a regularizer ρ to its diagonal elements, e.g., $\rho = 10^{-8}$ as in [7]. This ensures the convexity of the resulting QP problem.

1) Acceleration minimization: As done for instance in [8] and [25], we minimize the acceleration of the polynomials forming the trajectories using a cost of the form

$$\boldsymbol{\alpha}^T \int_0^{\Delta t} \ddot{\boldsymbol{\eta}}_n(\tau, 0)\, \ddot{\boldsymbol{\eta}}_n(\tau, 0)^T\, d\tau\, \boldsymbol{\alpha}, \quad (12)$$

where Δt denotes the segment duration. The integral has an analytic solution for all n-th order polynomials and can be computed offline as function of Δt.

2) Reference base state: To drive the robot toward the desired goal state, we penalize the deviation of the final base position, velocity, and acceleration from the ones of the reference trajectory. Furthermore, to enable tracking of the desired path with tunable accuracy, we sample the position deviation from the reference trajectory, which is also useful as regularization to counteract drift.

---

[5]Since the sequence of swing and stance segments is fixed prior to the optimization, the segment type is known from l. Accordingly, the definitions of $\boldsymbol{T}_{F,l}$ and $\boldsymbol{s}_{F,l}$ can be inferred from the value of their subscripts.

3) Hip to foot distance: To avoid solutions with excessively wide or narrow footprints, we penalize the distance between the hip and foot of a leg as

$$\|_I \boldsymbol{r}_{IB}^{xy}(t) + \boldsymbol{R}_{IB}^{xy}(t) \, _B \boldsymbol{r}_{BH}^{xy} - _I \boldsymbol{r}_{IF}^{xy}(t)\|_2^2, \quad (13)$$

where $\boldsymbol{r}_{BH}$ is the position from base to hip. We sample this objective a fixed number of times per stance segment.

4) ZMP and support polygon center: Regarding robust balancing and load distribution on the legs, it is desirable to keep the ZMP away from the edges and vertices of the support polygon. We thus penalize its distance from the geometric center of the support polygon by

$$\left\|_I \boldsymbol{r}_{IB}^{xy}(t) - \frac{1}{\ddot{z}_B(t) - g_z}\big(z_B(t)\,(_I \boldsymbol{a}_{IB}^{xy}(t) - _I \boldsymbol{g}^{xy}) + \boldsymbol{L}^{xy}(t)\big)\right.$$
$$\left. - \frac{1}{n_{\mathcal{G}}(t)} \sum_{F \in \mathcal{G}(t)} {}_I \boldsymbol{r}_{IF}^{xy}(t)\right\|_2^2, \quad (14)$$

where $\mathcal{G}(t)$ denotes the set of grounded legs at time t and $n_{\mathcal{G}}(t)$ their number.

C. Constraints

The following list gives an overview of the constraints that contribute to (5).

1) Initial states: We constrain the initial trajectory positions and velocities to match a fused state obtained from linear interpolation between the measured robot state and the previous solution. We do not constrain the initial base acceleration, because it decreases the reactivity against disturbances and could prevent fulfillment of the ZMP balance criterion.

2) Segment junctions: To obtain a twice continuously differentiable base trajectory, position, velocity, and acceleration of successive segments need to match at the junctions:

$$\begin{bmatrix} \boldsymbol{T}_B(t_{B,k+1}^{\text{start}}, t_{B,k}^{\text{start}}) & -\boldsymbol{T}_B(t_{B,k+1}^{\text{start}}, t_{B,k+1}^{\text{start}}) \\ \dot{\boldsymbol{T}}_B(t_{B,k+1}^{\text{start}}, t_{B,k}^{\text{start}}) & -\dot{\boldsymbol{T}}_B(t_{B,k+1}^{\text{start}}, t_{B,k+1}^{\text{start}}) \\ \ddot{\boldsymbol{T}}_B(t_{B,k+1}^{\text{start}}, t_{B,k}^{\text{start}}) & -\ddot{\boldsymbol{T}}_B(t_{B,k+1}^{\text{start}}, t_{B,k+1}^{\text{start}}) \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_{B,k} \\ \boldsymbol{s}_{B,k+1} \end{bmatrix} = \boldsymbol{0}.$$
$$(15)$$

However, we omit the constraint on the acceleration at junctions that mark the transition between two potentially disjoint support polygons[6] since, in such a case, the ZMP needs to be able to jump between the two[7].

We apply similar constraints on the position and velocity of successive feet segments to obtain once continuously differentiable trajectories. However, successive stance segments need special consideration in order not to introduce redundant constraints, as numerical optimization routines might require the equality constraint matrix to have full rank. We thus write the corresponding constraints as

$$\begin{bmatrix} \boldsymbol{T}_{F,l}(t_{F,l+1}^{\text{start}}, t_{F,l}^{\text{start}}) & -\boldsymbol{T}_{F,l}(t_{F,l+1}^{\text{start}}, t_{F,l+1}^{\text{start}}) \\ \boldsymbol{V}(t_{F,l+1}^{\text{start}}, t_{F,l}^{\text{start}}) & -\boldsymbol{V}(t_{F,l+1}^{\text{start}}, t_{F,l+1}^{\text{start}}) \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_{F,l} \\ \boldsymbol{s}_{F,l+1} \end{bmatrix} = \boldsymbol{0},$$
$$(16)$$

---

[6]For instance, the support polygons of a pacing gait with non-overlapping stance phases are disjoint. However, even for overlapping stance phases, the support polygons in the QP problem may be disjoint depending on the size of the margins chosen for the ZMP balance criterion constraint below.

[7]As noted in [7], discontinuities in the acceleration are undesirable from a controls perspective. However, they are not necessarily unphysical, in particular, if a hard contact model is assumed.

where $\boldsymbol{V}(t, t_{F,l}^{\text{start}}) = \begin{bmatrix} (t - t_{F,l}^{\text{start}})^2 & (t - t_{F,l}^{\text{start}}) & 1 & 0 & 0 \end{bmatrix}$. The third row of (16) requires the x-component of the velocities expressed in the foot frame to be equal. Since the y-component is zero by construction, see (10), it must not be constrained.

3) Leg extension: To prevent the legs from reaching kinematic limits, we require each foot to lie within a regular polygon centered at the hip position. We sample this inequality constraint at a fixed frequency.

4) ZMP balance criterion: The formulation of the ZMP balance criterion as proposed in Section II-B leads to inequality constraints on the base and stance feet of the form

$$_I \boldsymbol{e}^T \Big(_I \boldsymbol{r}_{IB}^{xy}(t) - \frac{1}{\ddot{z}_B(t) - g_z}\big(z_B(t)\,(_I \boldsymbol{a}_{IB}^{xy}(t) - _I \boldsymbol{g}^{xy})$$
$$+ \boldsymbol{L}^{xy}(t)\big) - _I \boldsymbol{r}_{IF}^{xy}(t)\Big) \le \epsilon, \quad (17)$$

where $\boldsymbol{e} \in \mathbb{R}^2$ is a normal vector describing the direction of an edge. The scalar ϵ can either be a positive relaxation margin for one- and two-leg support phases or a negative safety margin that prevents solutions where the ZMP lies near the support polygon boundaries. We sample the resulting constraints at a fixed frequency, except during full flight phases, as the ZMP does not exist during these.

5) Pure walking: If a solution should be found that keeps the stance feet in place, we add equality constraints that require the stance coefficients parameterizing the quadratic velocity polynomials to be zero, i.e., $\boldsymbol{\alpha}_{F,l} = \boldsymbol{0}$. This enables our TO to be used for conventional quadrupedal robots that have no wheels.

D. Warm Starting

In terms of optimality, using an initial guess is not required, since we solve a convex QP problem, which thus has a unique optimum. However, the number of iterations of numerical optimization routines can be significantly reduced by providing a suitable initial guess. We thus initialize the coefficients describing the initial state of each segment with the state of the solution from the previous optimization at the corresponding, shifted time instant $\hat{t}^{\text{start}}$ and set the remaining segment coefficients to zero. By "shifted" we mean that, for a segment start time t^{start}, the previous solution is evaluated at $\hat{t}^{\text{start}} = t^{\text{start}} + \Delta t_p$, where Δt_p is the time elapsed since the start of the previous optimization (we assume that internally the trajectories start at zero). By consequence, there is no solution available for the last Δt_p long portion of the trajectories. As a simple remedy, we set the associated segment coefficients to match the final state of the previous solution. In particular, for large position offsets, this is a better choice than setting them to zero.

E. Input Trajectories

As shown in Fig. 3, our optimization takes the angular trajectory components, the vertical base motion, and a planar reference base trajectory that should be tracked as inputs. These can be represented by any parameterization ensuring twice continuous differentiability. We choose sequences of

quintic polynomials with equality constraints similar to (15), which we shape using cost terms of the form (12) and equality constraints on junction positions, velocities, and/or accelerations. The latter are obtained by fusing information from the previous solution, the reference goal state, the measured state, and the gait pattern in the case of full flight phases. Namely, we require the vertical base acceleration to match gravity during these.

Since the resulting minimization problems contain linear equality constraints only, their solutions can directly be obtained by formulating the dual problems and solving the associated systems of linear equations, thus requiring no iterative optimization routines.

IV. RESULTS AND DISCUSSION

We have implemented our TO in C++. For matrix computations we use the open-source linear algebra library *Eigen* [26] and for solving (5) the open-source, state-of-the-art solver *OSQP* [27], which is based on the alternating direction method of multipliers and exploits the sparseness of our problem. Kinematics and dynamics computations are performed with the open-source libraries *Kindr* [28] and *RBDL* [29], and the simulations are carried out in the robot simulation environment *Gazebo* [30] with *ODE* [31] as physics engines.

A. Setup

We tested our framework in simulations of ANYmal equipped with actuated, non-steerable wheels[8]. Based on its joint configuration, we set the base pitch trajectory to zero in order to fulfill the condition stated in Section II. This ensures that the integral in (11) does not depend on any optimization variables. We further set the roll motion to zero, since we did not consider it essential to execute the motions we show in the following. For tracking of the trajectories, we used the WBC described in [24], that generates joint and wheel actuation torques at $400\,\text{Hz}$ while accounting for various constraints, such as actuator limitations and friction cone constraints.

Regarding solution complexity, we choose the maximal segment durations uniformly as $0.2\,\text{s}$, and enforce sampled objectives and constraints every $0.1\,\text{s}$, except for the ZMP balance criterion which we sample at $0.05\,\text{s}$. For a planning horizon of $2\,\text{s}$, this leads to QP problem sizes in the order of 6×10^2 optimization variables, 3×10^2 equality constraints, and 7×10^2 inequality constraints. The resulting total computation times of the TO are in the order of $20\,\text{ms}$, where solving the QP problem makes up roughly half of the time[9].

B. Simulations

Fig. 5 illustrates trajectories generated by our TO for different gaits and references. Thanks to the pure walking

[8]For realistic simulation, we use the full body dynamics, where the base, leg, and wheel inertia values are obtained from the computer-aided design (CAD) model of ANYmal. The actuator torque and velocity limits are considered as well.
[9]The times stated in this work and the video are always for the complete TO, i.e., they include the computation times of all blocks enclosed by the dotted rectangle in Fig. 3. All results were obtained on a 2.5 GHz quad-core Intel Core i7 laptop.
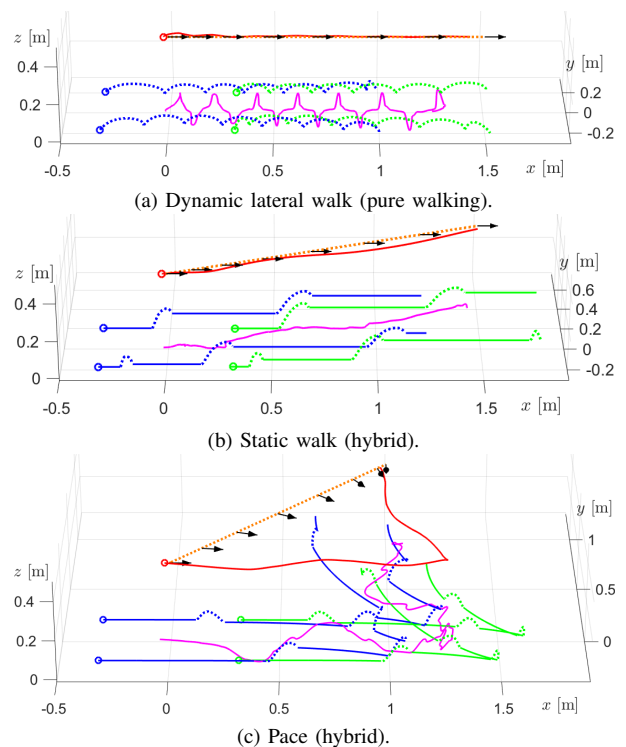

Fig. 5. Trajectories generated by our TO for different gaits and reference motions (dotted orange). The base trajectory is shown in red, the planned ZMP trajectory in magenta, and the fore and hind feet trajectories in green and blue, respectively, with the solid and dotted portions representing stance and swing phases, respectively. The initial feet and base positions are marked by circles. The yaw motion, which is a prescribed input, is shown as black arrows that indicate the robot's heading direction.

constraint in Section III-C5, our TO can be employed to generate motions for quadrupedal robots with point-contact feet, as shown in Fig. 5a for dynamic lateral walking (see Fig. 7 for an illustration of the gait patterns). However, the real strength of our approach is demonstrated in Fig. 5b and 5c, where driving considerably reduces the length and number of the steps required to reach the goal position. Remarkable is, in particular, the solution obtained for the reference in Fig. 5c. Instead of trying to directly reach the goal state by laterally pacing, our TO finds a solution similar to backwards parking with a car. In comparison, this considerably reduces lateral shifting of the base, which in our experience is crucial for robust execution of pacing gaits, since these inherently exhibit large lateral accelerations[10]. This result highlights the benefit of computing the planar components of the feet and base trajectories in a single optimization, as this behavior could not be expected by independently computing the feet trajectories using heuristics.

The reader is encouraged to watch the accompanying video[11], as it shows the execution of these motions and presents further results, including locomotion with a hybrid gait that conventional quadrupedal robots cannot execute.

Fig. 6 shows tracking of a reference trajectory with a full flight trotting gait. The motion is demanding to execute, as

[10]It should be emphasized that we did not impose a large cost on deviations from the sampled reference trajectory compared to the desired final state and the acceleration minimization. Note that we used the same weights in all presented simulations, including the video.
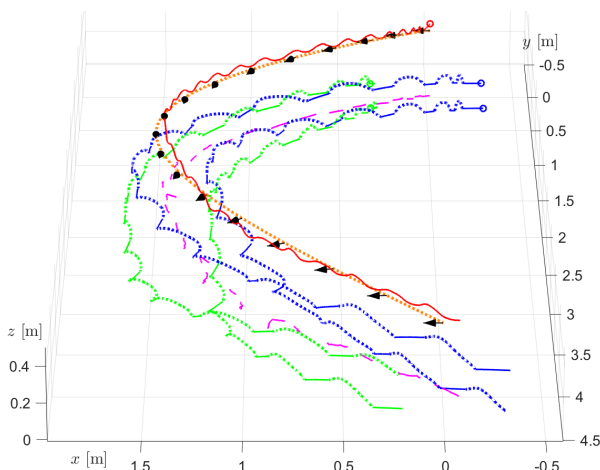[11]Also available at https://youtu.be/I1aTCTc0J4U.

Fig. 6. Tracking of a reference trajectory in simulation with our TO running in receding horizon fashion and the planning horizon set to $2\,\mathrm{s}$. The gait is trotting with full flight phases. Shown are the continuously updated setpoints, that is, the positions that were to be tracked by the WBC. The choice of colors and symbols is the same as in Fig 5.
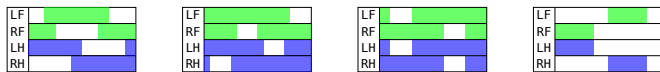

Fig. 7. Gait patterns of the motions shown in the previous figures. From left to right, with the stride durations in brackets: Dynamic lateral walk ($0.6\,\mathrm{s}$), static walk ($1.7\,\mathrm{s}$), pace ($1.0\,\mathrm{s}$), flying trot ($0.6\,\mathrm{s}$). Swing and stance phases are shown in white and color, respectively. Note that the time scales are normalized w.r.t. the stride durations.

it does not only contain full flight phases but also requires the robot to drive a curve and even change its orientation by $90°$ while moving forward. However, by running our TO in receding horizon fashion[12], we are able to follow the reference and compensate for the various sources of disturbances, such as the simplified robot model, computation delays, and the accumulation of tracking errors. For the chosen planning horizon of $2\,\mathrm{s}$, the update frequency was measured to be in the order of $50\text{-}60\,\mathrm{Hz}$. We furthermore found that not using the warm starting scheme of Section III-D increases the QP solver time by up to $40\,\%$.

C. Discussion of real-world applicability

As indicated by the simulation results, our TO is a promising candidate for application on real robots, since it can be run online at high frequency in receding horizon fashion and generates a rich set of motions. Conceptually, it is closely related to the ZMP-based approaches [6]–[8], which have demonstrated impressive performance on hardware. In particular, the implementations of [7], [8] successfully run on ANYmal, despite using the imperfect on-board state estimation [32] and the TOs not accounting for the leg inertias, which are relatively large due to the knee actuators. From our experience with [7], [8], we expect that the main step required to bring our TO from simulation to real hardware will be related to handling the drift in the state estimation and accounting for unperceived terrain irregularities. A simple and effective measure for these two purposes consists of always computing the trajectories w.r.t. the currently estimated ground plane, see the *plan frame* in

[8] for details. In combination with the compliant behavior of the WBC [33], this should allow to blindly traverse significant irregularities, as demonstrated for steps and inclines in [24].

Further measures that we found to be important for robustness are accounting for early and late touchdowns and reacting to unexpected loss or gain of contact. These issues can be addressed by an online adaptation of the vertical feet trajectories and the gait pattern. Note that recomputation of the vertical feet trajectories can be done instantaneously since it is decoupled from the QP problem.

V. CONCLUSION AND OUTLOOK

We have shown how trajectories for wheeled-legged quadrupedal robots can be generated, such that complex motions requiring simultaneous driving and stepping can be executed. By solving for angular, vertical, and planar trajectory components in a cascaded fashion and using a linear formulation of the ZMP balance criterion, we rely on QP only. Nevertheless, we are able to execute gaits with full flight phases. In addition, our implementation meets and, depending on the motion, even beats the computation speed of competing TOs for quadrupedal robots, despite solving a problem that due to the motion of the wheels in stance phase is more complex.

In the future, we plan to investigate how much the condition of zero pitch can be relaxed for ANYmal, in order to execute gaits such as bounding, which would be useful to cross obstacles and steps at high driving speed. Further, we would like to generate the gait pattern online as function of the reference trajectory, terrain irregularities, and obstacles such as to ensure robust dynamic balancing and to minimize the number of steps in favor of energy consumption.

APPENDIX

ANYmal: Condition for Linear Rolling Constraint

For ANYmal, it holds that under the condition of zero base pitch angle w.r.t. the ground plane, the wheel rotation axes and the ground projected x-axis of the base frame – which we define as the *heading direction* – are always perpendicular. According to our definition of the frame F (see Fig. 2), this implies that the x-axis of F is parallel to the heading direction. Since we have from the rolling constraint that the velocity of F must be parallel to its x-axis, it follows that the rolling constraint is independent of the relative position of F w.r.t. the base. To prove this claim, we thus verify in the following that the wheel axes and the heading direction are indeed perpendicular for zero base pitch.

To this end, we parameterize the base orientation by an intrinsic z-y'-x'' sequence of Euler angles[13]:

$$\boldsymbol{R}_{IB} = \boldsymbol{R}_z(\psi)\boldsymbol{R}_y(\chi)\boldsymbol{R}_x(\phi), \tag{18}$$

where ψ, χ, ϕ are the base yaw, pitch, roll angles, respectively, and where $\boldsymbol{R}_a(\cdot)$ denotes the rotation matrix with rotation axis a. The joint configuration of ANYmal's legs corresponds to

---

[12]In our context, this means that we continuously update the trajectories by rerunning the TO. Therefore, each set of trajectories is only tracked as long as it takes to compute a new one.

[13]We do not provide any intermediate results due to the corresponding terms being very long and of limited interest. However, the correctness of our findings can easily be verified with a symbolic computation tool.

an intrinsic x-y'-y" sequence of Euler angles, which describes the orientation from base to shank S^{14}:

$$R_{BS} = R_x(\alpha)R_y(\beta)R_y(\gamma), \tag{19}$$

where α, β, γ are the hip abduction/adduction, hip flexion/extension, knee flexion/extension angles, respectively. Note that the shank frame does not coincide with the foot frame F, but that the projections of their y-axes onto the ground plane are parallel. The orientation from the inertial frame to the shank frame is given by

$$R_{IS} = R_{IB}R_{BS}. \tag{20}$$

Accordingly, the direction of the wheel axis is given by the second column of R_{IS}, and the heading direction by

$$_I h_B = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \end{bmatrix}^T. \tag{21}$$

As can be shown, their dot product evaluates to

$$\sin(\phi + \alpha)\sin(\chi). \tag{22}$$

Thus, it follows that the wheel axis is perpendicular to the robot's heading direction for a base pitch angle equal to zero.

REFERENCES

[1] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Bloesch, *et al.*, "ANYmal - toward legged robots for harsh environments," *Advanced Robotics*, 2017.

[2] G. Bledt, P. M. Wensing, and S. Kim, "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the MIT Cheetah," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 4102–4109.

[3] M. H. Raibert, *Legged robots that balance*. MIT press, 1986.

[4] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, July 2018.

[5] A. W. Winkler, "Tutorial: Gait and trajectory optimization for legged robots," June 2018. [Online]. Available: https://www.youtube.com/watch?v=KhWuLvb934g

[6] A. W. Winkler, F. Farshidian, D. Pardo, M. Neunert, and J. Buchli, "Fast trajectory optimization for legged robots using vertex-based ZMP constraints," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2201–2208, Oct 2017.

[7] C. D. Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic locomotion and whole-body control for quadrupedal robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 3359–3365.

[8] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2261–2268, July 2018.

[9] M. Vukobratović and B. Borovac, "Zero-moment point – thirty five years of its life," *International journal of humanoid robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[10] P. Sardain and G. Bessonnet, "Forces acting on a biped robot. Center of pressure – zero moment point," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 34, no. 5, pp. 630–637, Sept 2004.

[11] B. H. Wilcox, "ATHLETE: A limbed vehicle for solar system exploration," in *2012 IEEE Aerospace Conference*, March 2012, pp. 1–9.

[12] W. Reid, F. J. Prez-Grau, A. H. Gktoan, and S. Sukkarieh, "Actively articulated suspension for a wheel-on-leg rover operating on a martian analog surface," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 5596–5602.

[13] F. Cordes, A. Babu, and F. Kirchner, "Static force distribution and orientation control for a rover with an actively articulated suspension system," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 5219–5224.

[14] K. Hashimoto, T. Hosobata, Y. Sugahara, Y. Mikuriya, H. Sunazuka, M. Kawase, H. ok Lim, and A. Takanishi, "Realization by biped leg-wheeled robot of biped walking and wheel-driven locomotion," in *2005 IEEE International Conference on Robotics and Automation (ICRA)*, April 2005, pp. 2970–2975.

[15] J. Lim, I. Lee, I. Shim, H. Jung, H. M. Joe, H. Bae, O. Sim, J. Oh, T. Jung, S. Shin, K. Joo, M. Kim, K. Lee, Y. Bok, D.-G. Choi, B. Cho, S. Kim, J. Heo, I. Kim, J. Lee, I. S. Kwon, and J.-H. Oh, "Robot system of DRC-HUBO+ and control strategy of team KAIST in DARPA robotics challenge finals," *Journal of Field Robotics*, vol. 34, no. 4, pp. 802–829.

[16] T. Klamt and S. Behnke, "Planning hybrid driving-stepping locomotion on multiple levels of abstraction," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1695–1702.

[17] T. Klamt, D. Rodriguez, M. Schwarz, C. Lenz, D. Pavlichenko, D. Droeschel, and S. Behnke, "Supervised autonomous locomotion and manipulation for disaster response with a centaur-like robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018.

[18] J. A. Smith, I. Sharf, and M. Trentini, "PAW: a hybrid wheeled-leg robot," in *2006 IEEE International Conference on Robotics and Automation (ICRA)*, May 2006, pp. 4043–4048.

[19] P. R. Giordano, M. Fuchs, A. Albu-Schaffer, and G. Hirzinger, "On the kinematic modeling and control of a mobile platform equipped with steering wheels and movable legs," in *2009 IEEE International Conference on Robotics and Automation (ICRA)*, May 2009, pp. 4080–4087.

[20] M. Giftthaler, F. Farshidian, T. Sandy, L. Stadelmann, and J. Buchli, "Efficient kinematic planning for mobile manipulators with non-holonomic constraints using optimal control," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 3411–3417.

[21] T. Tanaka and S. Hirose, "Development of leg-wheel hybrid quadruped "AirHopper" design of powerful light-weight leg with wheel," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2008, pp. 3890–3895.

[22] Boston Dynamics. Introducing Handle. Youtube. [Online]. Available: https://www.youtube.com/watch?v=-7xvqQeoA8c

[23] M. Geilinger, R. Poranne, R. Desai, B. Thomaszewski, and S. Coros, "Skaterbots: Optimization-based design and motion synthesis for robotic creatures with legs and wheels," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 160:1–160:12, July 2018.

[24] M. Bjelonic, C. D. Bellicoso, Y. de Viragh, D. Sako, F. Tresoldi, F. Jenelten, and M. Hutter, "Keep rollin' – whole-body motion control and planning for wheeled quadrupedal robots," *under review for IEEE Robotics and Automation Letters*, 2019.

[25] A. W. Winkler, C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5148–5154.

[26] G. Guennebaud, B. Jacob, *et al.*, "Eigen v3." [Online]. Available: http://eigen.tuxfamily.org

[27] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *arXiv:1711.08013*, Nov. 2017.

[28] C. Gehring, C. D. Bellicoso, M. Bloesch, R. Diethelm, P. Fankhauser, P. Furgale, M. Neunert, and H. Sommer, "Kindr – kinematics and dynamics for robotics." [Online]. Available: https://github.com/ANYbotics/kindr

[29] M. L. Felis, "RBDL: An efficient rigid-body dynamics library using recursive algorithms," *Autonomous Robots*, vol. 41, no. 2, pp. 495–511, Feb 2017.

[30] Open Source Robotics Foundation, "Gazebo." [Online]. Available: http://gazebosim.org

[31] S. Russell, "ODE – open dynamics engine." [Online]. Available: http://www.ode.org

[32] M. Bloesch, M. Hutter, M. Hoepflinger, S. Leutenegger, C. Gehring, C. David Remy, and R. Siegwart, "State estimation for legged robots - consistent fusion of leg kinematics and IMU," July 2012.

[33] C. D. Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov 2016, pp. 558–564.

---

[14]The shank is the rigid body connecting the knee and wheel joints. Note that for ANYmal their rotation axes are parallel. Given the definition of R_{BS}, the y-axis of the shank frame is thus parallel as well.